

# Designing and implementing an smart monitoring and management system of environmental and electrical conditions of server rooms



University of Guilan

Mokadar Daemdoost, Amin  
amindaemdoost@yahoo.com

Abbasi, Poorya  
hey@pooryaa.com

30 Jan 2023

## **Abstract**

The design and implementation of a server room monitoring system, which tracks significant environmental factors like temperature, humidity, and dust levels, are presented in this thesis. The system is made to give real-time updates on the circumstances in the server room, ensuring that the servers and other equipment are working properly.

In order to retrieve data from the database and return it in a format that the front-end can use, Node.js was used to build a REST API for the back-end. We used Prisma and PostgreSQL to provide the necessary data management and storage features. The efficient and scalable handling of massive amounts of data is ensured by the use of Node.js, Prisma, and PostgreSQL, and the type-safe API offered by Prisma makes it simpler to retrieve and manipulate the data.

An interface that provides real-time updates on the circumstances in the server room was made for the front-end using React, Next.js, and Tailwind. While Next.js offers server-side rendering and enhanced performance, React offers a robust and adaptable framework for creating user interfaces. Rapid and effective front-end development is made possible by Tailwind's selection of pre-made UI elements and styling tools.

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Backgrounds . . . . .	3
1.1.1	Monitoring the environment . . . . .	3
1.1.2	Security . . . . .	5
1.1.3	Real-time monitoring . . . . .	5
1.1.4	Data Visualization . . . . .	5
1.2	Purpose . . . . .	6
<b>2</b>	<b>Data Generator Selection and Implementation</b>	<b>7</b>
2.1	Introduction . . . . .	7
2.2	Methodology . . . . .	7
2.3	Results . . . . .	7
2.3.1	Generated temperature sensor data . . . . .	7
2.3.2	Generated humidity sensor data . . . . .	9
2.3.3	Generated dust sensor data . . . . .	9
2.3.4	Generated electricity voltage sensor data . . . . .	9
2.3.5	Generated electricity current sensor data . . . . .	11
<b>3</b>	<b>Panel Design and Implementation</b>	<b>12</b>
3.1	Introduction . . . . .	12
3.2	Back-end . . . . .	12
3.2.1	Node js . . . . .	12
3.2.2	Postgresql . . . . .	13
3.2.3	Prisma . . . . .	13
3.3	Front-end . . . . .	13
3.3.1	React . . . . .	13
3.3.2	Next js . . . . .	14
3.3.3	Tailwind . . . . .	14
3.4	Panel overview . . . . .	14
3.4.1	Dashboard . . . . .	14
3.4.2	Sensors . . . . .	15

# Chapter 1

## Introduction

### 1.1 Backgrounds

Server rooms are a critical component of today's technology-driven world, with the growing reliance on and need for technological devices, ensuring that server rooms function properly has become an essential part of our daily lives. Server rooms are important to an organization because they contain infrastructure and critical equipment. Monitoring various parameters such as temperature, humidity, electricity and others helps ensure that the system is running smoothly and the first step in monitoring server rooms is to understand the different parameters that need to be monitored.

#### 1.1.1 Monitoring the environment

A monitoring system should be able to monitor the environment of the server room and alert the administrators in case of any abnormality. The following list contains some of the most important parameters that should be monitored in a server room.

- Temperature

Server room temperature should be maintained at a stable, suitable level to ensure the proper functioning of equipment. The recommended range of temperatures is between 18°C and 27°C. [1]

**Too high temperature:** If the temperature inside a server room rises above the recommended range, it can cause several problems. High temperatures can make equipment operate less efficiently and potentially fail altogether. High temperatures can also decrease the life expectancy of electronic devices and increase the chance that stored data will be corrupted. [2]

**Too low temperature:** If the temperature in a server room is allowed to fall below the recommended range, it can cause several issues. The cold air can lead to condensation—which leads directly to corrosion and equipment damage. In addition, low temperatures can decrease the efficiency of equipment and make it more likely to fail. [2]

- Humidity

On the other hand, High humidity levels can lead to condensation, which causes corrosion of the equipment and short circuits. High humidity levels can also create conditions that are conducive to the growth of mold and other microorganisms, which in turn damage equipment and affect indoor air quality. [3]

- Dust

Dust accumulation in a server room can degrade the performance and lifespan of equipment. Dust can block air vents, causing overheating, and it also attracts moisture leading to corrosion or other problems. So monitoring the levels of dust in a server room can help identifying and addressing potential problems.

The suitable range for dust density in a server room is around 0 to 3mg/m<sup>3</sup>.

- Water Leakage

Serious consequences can result from water leakage in a server room, even if only a small amount of water leaks onto equipment. It is important to have proper monitoring systems and alerts that will notify personnel as soon as possible after any leak occurs. [4]

- Electricity

Monitoring the state of electricity, including voltage and current, is important in a server room to ensure the stability and reliability of the power supply to the equipment. Electrical voltage and current fluctuations can lead to problems with electronic equipment, such as data loss and corruption. In order to minimize these risks, server rooms are typically equipped with uninterruptible power supplies (UPS) and surge protection devices that help stabilize the voltage. The following list contains some common ranges for voltage and current.[4]

**Voltage:** The recommended operating range in from 208V to 240V and the maximum recommended limit is 264V.

**Current:** The recommended operating range in from 20A to 40A (per phase)

- Movement

Movement sensors, also known as motion detectors, can be used in order to detect unauthorized access to the room by detecting movements within the room and alerting the administrators.

- Smoke

Smoke sensors can be used to detect smoke and alert the administrators in case of a fire.

- Authorization

Authorization devices, such as cameras, fingerprints, keypads, etc., can be used to detect unauthorized access to the room by detecting the presence of a specific person.

### 1.1.2 Security

Another aspect of a monitoring system is that it should be able to prevent any intrusions and ensure that data remains intact and that access is restricted to only those with the appropriate privileges.

- Confidentiality And Authentication

in every IOT system, a proper authentication is crucial to ensure the security of the data so it wouldn't get into the wrong hands or get mutated by malicious actors and because the data is being transmitted remotely, more security measures has to be taken. [5, 6]

- Authorization

Security and reliability are the most important factors in our use case of network and other security measure implementations. There are trade offs like losing flexibility but these are the trade offs that we are willing to make to reach the maximum level of security possible.[5, 6]

- Integrity

In an IOT system, there is a risk of intruders attempting to impersonate legitimate users and alter or manipulate data. So it is important to implement verification and encryption methods to make sure in an event of a breach, the integrity of the data is still ensured and no nefarious actor in the system can mutate or access the data in any way because the consequences can be quite catastrophic. [5, 6]

### 1.1.3 Real-time monitoring

Real-time monitoring is important in a monitoring system to ensure that the administrators are notified as soon as possible in case of any abnormality.

### 1.1.4 Data Visualization

Data visualization is an important part of a monitoring system, because it can provide a clear and compact representation of the data sent by the sensors. This allows administrators to quickly and easily identify trends, patterns, and anomalies in the server room environment, such as changes in temperature, humidity, power, and other environmental factors.

Data visualization can be achieved through a variety of means, such as graphs, charts, and maps. These visual representations can provide a lot of information in real-time, allowing administrators to make informed decisions about the state of their server room and respond quickly to any issues or problems. For example, graphs can be used to track trends over time.

Table 1.1: Electrical sensors

Name	Measurement Unit	Output
Voltage Sensor	Voltage	ADC
Current Sensor	Amper	ADC

Table 1.2: Environment sensors

Name	Measurement Unit	Operating temp	Desired values
Temperature sensor	Celsius	-20°C - 50°C	18°C - 27°C
Smoke sensor	-	-20°C - 50°C	-
Humidity sensor	Percent	-20°C - 50°C	40% - 60%
Water Leakage Sensor	-	-20°C - 50°C	-
Dust sensor	-	-20°C - 50°C	-

## 1.2 Purpose

The purpose of this thesis is to demonstrate how an effective and efficient monitoring system can be implemented for server rooms using sensors, which are introduced in three categories in the tables 1.1, 1.2 and 1.3. The system will be designed to provide real-time data to a website, so that it can be monitored anywhere with internet access. This thesis will examine how this system was designed and implemented, including the selection of sensors and development of an online panel for monitoring data visualization.

Table 1.3: Security sensors

Name
Movement sensor
Fingerprint sensor
Camera

# Chapter 2

## Data Generator Selection and Implementation

### 2.1 Introduction

In this implementation, in order to demonstrate how the monitoring system works instead of using actual sensors, a data generator has been used and it simulates reading of sensors (such as humidity, temperature and electricity).

### 2.2 Methodology

A data generator should have a random yet realistic pattern for effective system testing, so just using a simple random number would be insufficient. Therefore, in this thesis a controlled version of choosing a random number was used to mimic some parts of real-world criteria and avoid sudden changes. The methodology for this process can be seen in the accompanying flowchart 2.1.

### 2.3 Results

As discussed in the methodology section, the program was implemented, and the generated values tested with different parameters was saved and reported as line charts in the subsequent sections.

#### 2.3.1 Generated temperature sensor data

- Parameters

**Starting Temperature sensor Value:** 22

**Minimum Possible Temperature sensor value:** 18

**Maximum Possible Temperature sensor value:** 27

**Maximum Possible Change in an iteration:** 0.5

- Results can be seen in figure 2.2



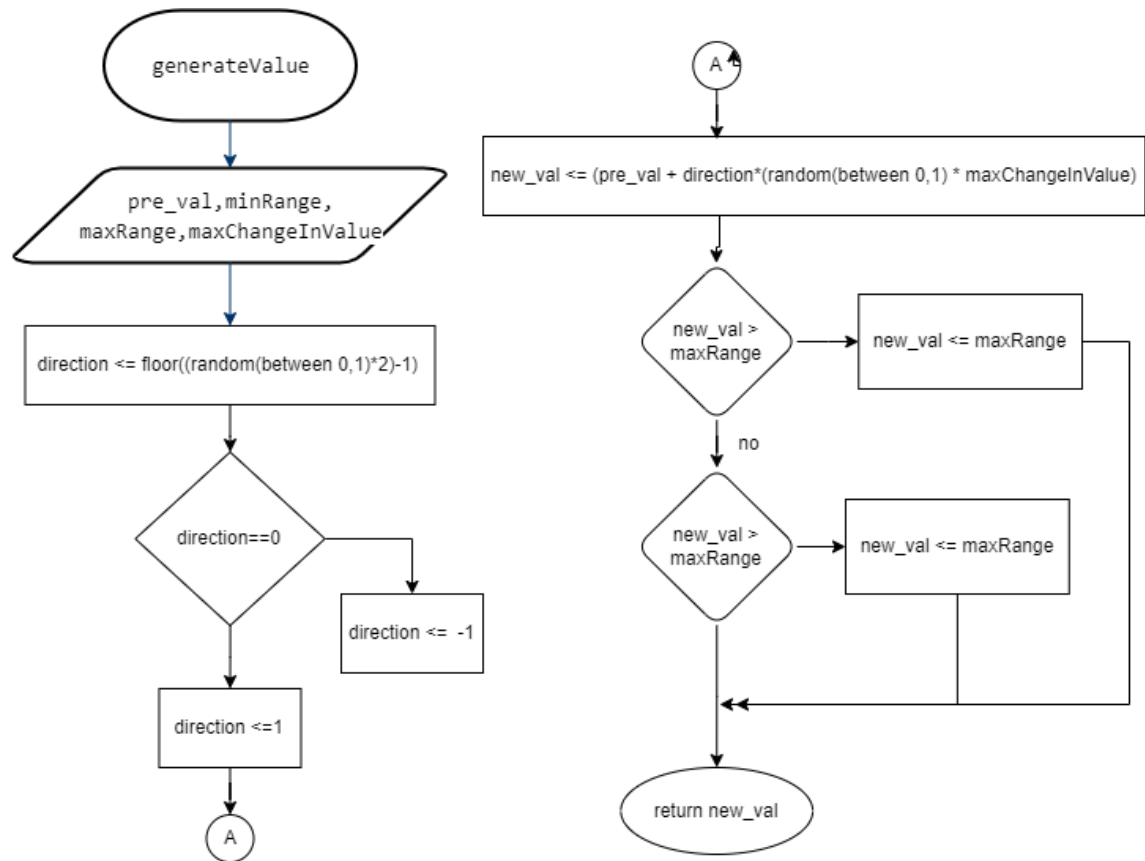
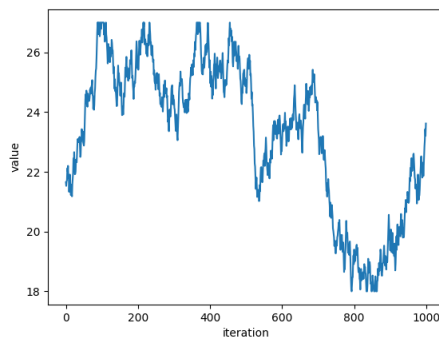
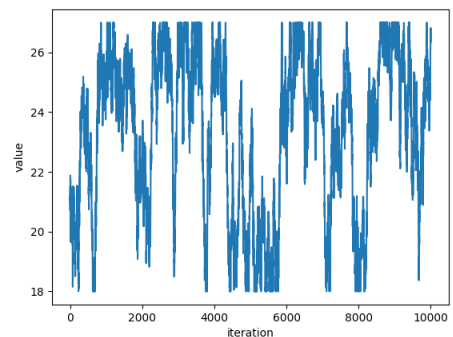


Figure 2.1: Flowchart of data generator



(a) For 1000 iterations



(b) For 10000 iterations

Figure 2.2: Sample generated temperature sensor data

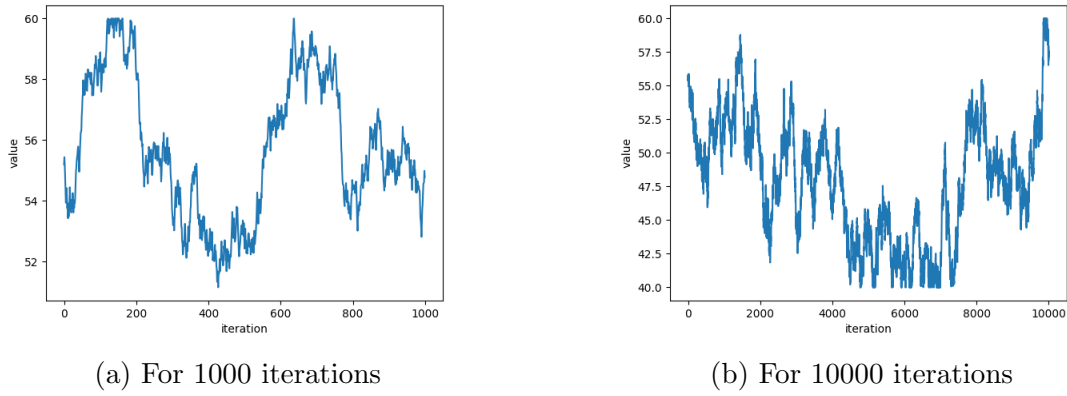


Figure 2.3: Sample generated humidity sensor data

### 2.3.2 Generated humidity sensor data

- Parameters

Starting humidity sensor value: 55

Minimum possible humidity sensor value: 40

Maximum possible humidity: 60

Maximum possible Change in an iteration: 0.5

- Results can be seen in figure 2.3

### 2.3.3 Generated dust sensor data

- Parameters

Starting dust sensor Value: 220

Minimum possible dust sensor value: 208

Maximum possible electricity dust sensor value: 264

Maximum possible change in an iteration: 1

- Results can be seen in figure 2.4

### 2.3.4 Generated electricity voltage sensor data

- Parameters

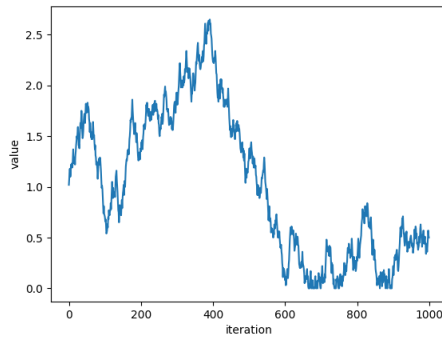
Starting electricity voltage value: 220

Minimum possible electricity voltage: 208

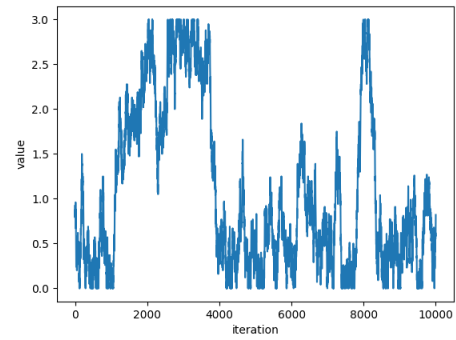
Maximum possible electricity voltage: 264

Maximum possible electricity change in an iteration: 1

- Results can be seen in figure 2.6

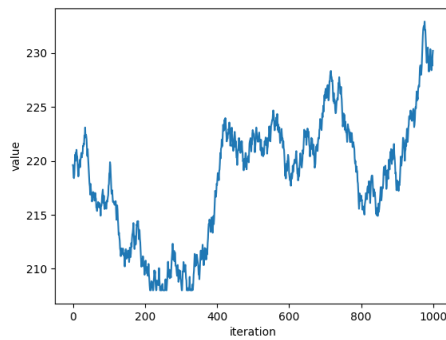


(a) For 1000 iterations

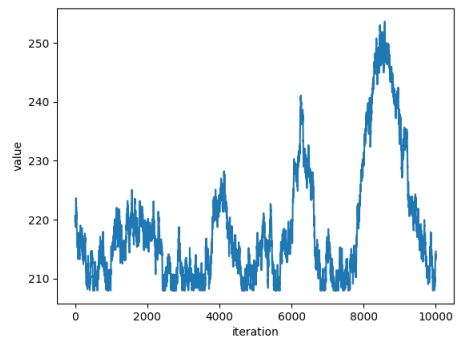


(b) For 10000 iterations

Figure 2.4: Sample generated dust sensor data

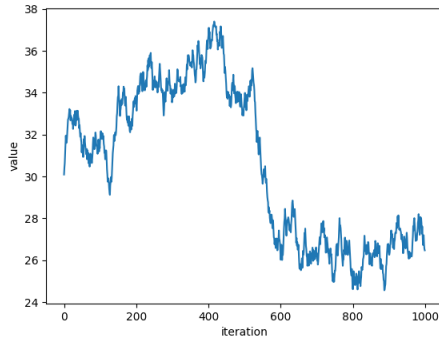


(a) For 1000 iterations

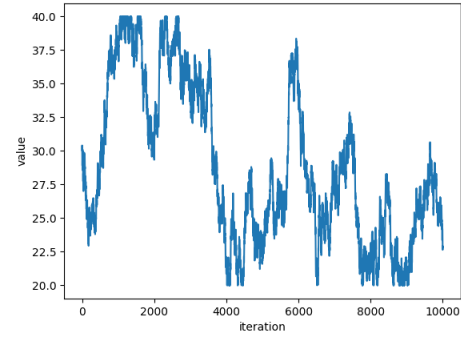


(b) For 10000 iterations

Figure 2.5: Sample generated voltage sensor data



(a) For 1000 iterations



(b) For 10000 iterations

Figure 2.6: Sample generated voltage sensor data

### 2.3.5 Generated electricity current sensor data

- Parameters

**Starting electricity Current Value:** 220

**Minimum possible electricity Current:** 208

**Maximum possible electricity Current:** 264

**Maximum possible change in an iteration:** 1

- Results can be seen in figure 2.6

# Chapter 3

## Panel Design and Implementation

### 3.1 Introduction

A good server room monitoring system panel is an essential component for ensuring the proper functioning of a server room environment. It provides real-time monitoring of key physical and environmental factors such as temperature, humidity, power, and network connectivity. The panel alerts administrators in case of any deviations from the defined thresholds, helping prevent potential data loss, hardware damage, and downtime. Additionally, a good server room monitoring system panel provides historical data and reporting capabilities, allowing administrators to track trends and make informed decisions about capacity planning and infrastructure improvements. With the right server room monitoring system panel in place, organizations can ensure the safety and reliability of their critical IT assets.

### 3.2 Back-end

Node.js, PostgreSQL, and Prisma can be used together to build a robust and scalable back-end for a server room monitoring panel.

#### 3.2.1 Node js

Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine, and is a critical component in building a server room monitoring panel's back-end. Node.js allows developers to use JavaScript on the server-side, providing a unified language for both client-side and server-side development. This helps to simplify the development process and increase developer productivity.

In a server room monitoring panel's back-end, Node.js can be used to create APIs or other server-side components that provide data for the monitoring panel. For example, it can be used to create a REST API that retrieves data from a database or other source and returns it in a format that can be consumed by the monitoring panel. Node.js is also well-suited for building fast and scalable applications, making it a good choice for a server room monitoring panel's back-end. Its asynchronous, non-blocking I/O model ensures that the back-end can handle a large number of concurrent requests without becoming slow or unresponsive. [7]

### 3.2.2 Postgresql

PostgreSQL is an open-source relational database management system (RDBMS) and is a critical component in building the back-end for a server room monitoring panel. PostgreSQL is a powerful and reliable database system that can be used to store the data collected by the monitoring panel, such as temperature, humidity, power, and network connectivity. It provides a range of features for managing and querying data, including support for SQL, transactions, and advanced data types. The use of a relational database, such as PostgreSQL, is important for a server room monitoring panel because it ensures the data is stored in a structured manner. This makes it easier to retrieve and analyze the data, and also provides a robust solution for managing the data over time, ensuring its reliability and longevity. PostgreSQL also provides security features, such as user authentication and access control, to ensure that the data stored in the database is protected. This is especially important for a server room monitoring panel where sensitive information, such as server uptime, may be stored.[8]

### 3.2.3 Prisma

Prisma is a modern database toolkit that provides a high-level API for working with databases and is a critical component in building the back-end for a server room monitoring panel. Prisma allows developers to work with databases, such as PostgreSQL, in a type-safe and efficient manner. It provides a set of TypeScript types for the data stored in the database, making it easy to catch bugs and errors at compile time. This helps to ensure the quality and reliability of the data stored in the database.

Prisma also provides a query builder for constructing and executing SQL queries, allowing developers to retrieve the data necessary for the monitoring panel. This makes it easy to retrieve and manipulate the data stored in the database, while also providing the performance and efficiency necessary for a server room monitoring panel's back-end. Prisma also provides a streamlined and modern API for working with databases, making it easier for developers to build robust and scalable back-ends. This can help to reduce development time and increase developer productivity.[9]

## 3.3 Front-end

### 3.3.1 React

React is a JavaScript library for building user interfaces, and is a critical component in building a server room monitoring panel. React allows for the creation of reusable components, making it easier to manage and maintain the codebase as the application grows in complexity. This helps to ensure that the monitoring panel is both scalable and flexible, allowing for new features and functionality to be added as needed.

In a server room monitoring panel, React can be used to create components for displaying real-time data, such as temperature, humidity, power, and network connectivity. This data can be fetched from APIs or other sources and dynamically

updated using React's state management. This helps to ensure that the monitoring panel provides an up-to-date and accurate representation of the server room environment at all times. React's hooks and context also make it easy to add alerts and notifications to the monitoring panel, ensuring that administrators are promptly notified of any deviations from the defined thresholds. This helps to prevent potential data loss, hardware damage, and downtime. [10]

### 3.3.2 Next js

Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine, and is a critical component in building a server room monitoring panel. Node.js allows developers to use JavaScript on the server-side, providing a unified language for both client-side and server-side development. This helps to simplify the development process and increase developer productivity.

In a server room monitoring panel, Node.js can be used to create APIs or other server-side components that provide data for the monitoring panel. For example, it can be used to create a REST API that retrieves data from a database or other source and returns it in a format that can be consumed by the monitoring panel.

Node.js is also well-suited for building fast and scalable applications, making it a good choice for a server room monitoring panel. Its asynchronous, non-blocking I/O model ensures that the monitoring panel can handle a large number of concurrent requests without becoming slow or unresponsive.[11]

### 3.3.3 Tailwind

Tailwind is a utility-first CSS framework, and is a critical component in building a server room monitoring panel. Tailwind provides a set of pre-defined CSS classes for quickly building responsive and customizable designs. This helps to significantly speed up the development process, as developers do not need to write custom CSS for common design elements.

In a server room monitoring panel, Tailwind can be used to style the components created with React. It provides a wide range of classes for controlling layout, typography, colors, and more, making it easy to create a consistent and professional look and feel for the monitoring panel. Tailwind also provides a set of responsive design classes, allowing the monitoring panel to adjust its layout and appearance based on the size of the user's screen. This helps to ensure that the monitoring panel is usable and accessible on a variety of devices, including desktop computers, laptops, tablets, and smartphones.[12]

## 3.4 Panel overview

### 3.4.1 Dashboard

The dashboard is the main page of the monitoring panel, and provides a high-level overview of the server room environment. In this state of the panel as it can be seen in the image 3.1, online sensors and their data distribution has been reported.

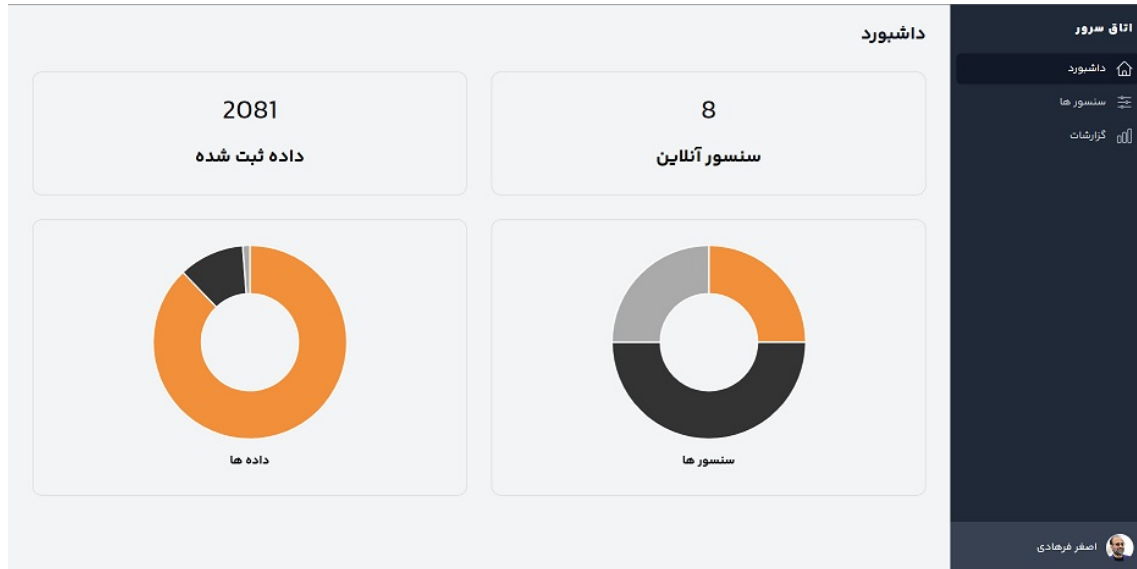


Figure 3.1: Dashboard tab in the panel

Figure 3.2 shows the Sensors tab in the panel. It displays a table with 9 sensors. The table has three columns: 'عنوان سنسور' (Sensor Title), 'نوع سنسور' (Sensor Type), and 'نامایش ویرایش' (Edit/View). The right sidebar contains navigation links: 'اتاق سرور' (Server Room), 'داشبورد' (Dashboard), 'سنسور ها' (Sensors), and 'گزارشات' (Reports).

عنوان سنسور	نوع سنسور	نامایش ویرایش
سنسور ولتاژ رک اصلی	الکتریکی	نمایش ویرایش
سنسور جریان رک اصلی	الکتریکی	نمایش ویرایش
سنسور رطوبت نزدیک درب	شرایط	نمایش ویرایش
سنسور دمای رک اصلی	شرایط	نمایش ویرایش
سنسور حرکت نزدیک درب	رخداد	نمایش ویرایش
سنسور نشت آب سراسری	رخداد	نمایش ویرایش
سنسور گرد و خاک سراسری	شرایط	نمایش ویرایش
سنسور دمای اصلی	شرایط	نمایش ویرایش

Figure 3.2: Sensors tab in the panel

### 3.4.2 Sensors

The sensors overview page provides a detailed view of the sensors in the server room and as it can be seen in the image 3.2, user can edit the sensor information or view the sensor data.

#### view sensor data

The sensor data page provides a detailed view of the sensor data and as it can be seen in the images 3.3, 3.4 and 3.5, user can view the sensor data in a graph.

#### sensor settings

In the sensor settings page, user can edit the sensor information and as it can be seen in the image 3.6, user can edit the sensor's name.



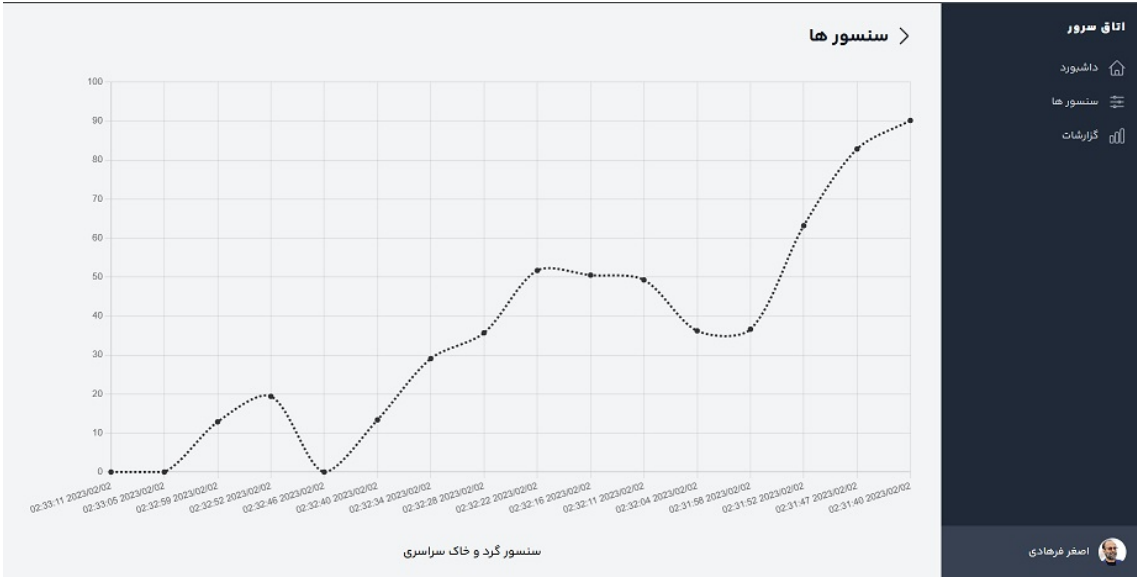


Figure 3.3: A dust sensor data in the panel

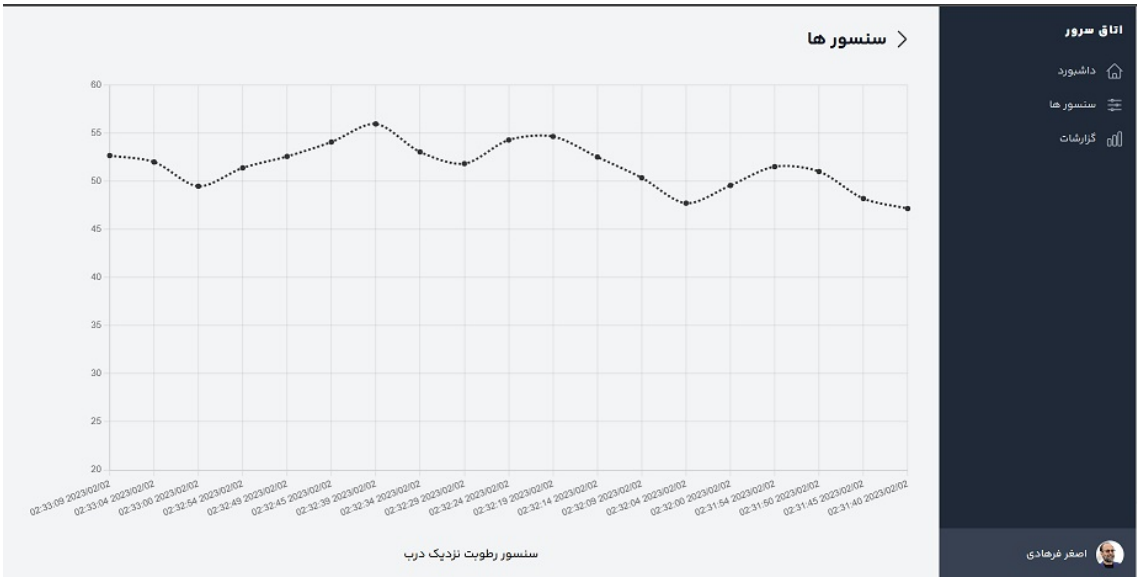


Figure 3.4: Humidity tab in the panel

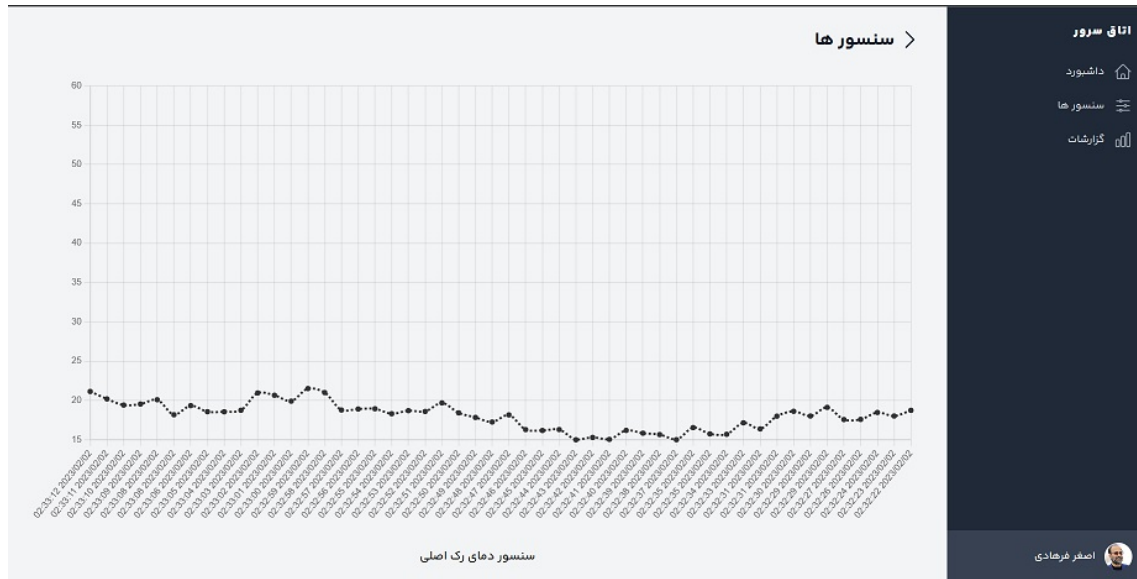


Figure 3.5: Temperature tab in the panel

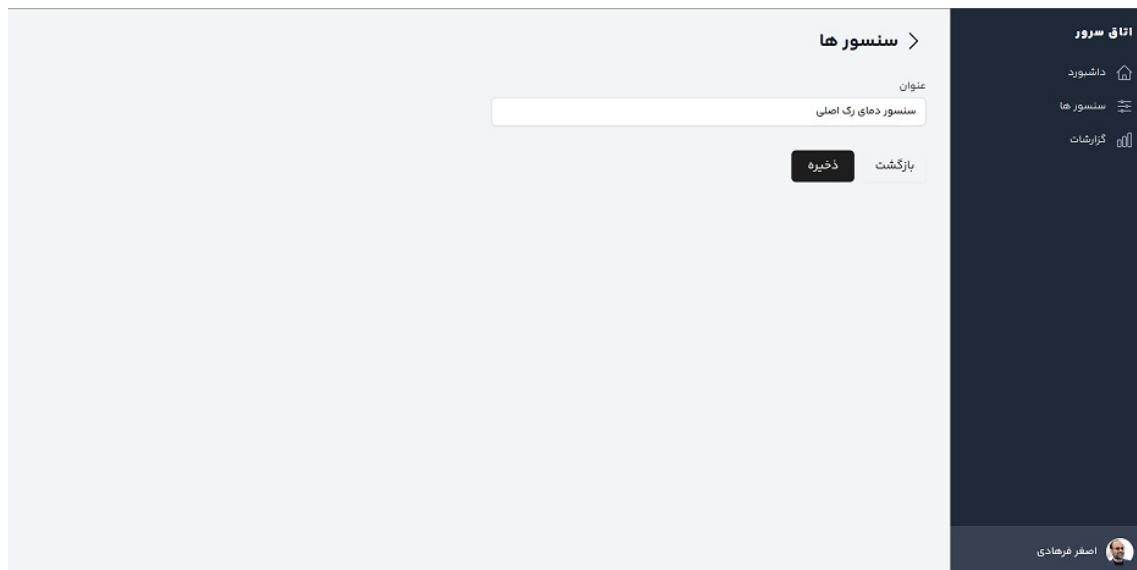


Figure 3.6: Sensor settings tab in the panel

# Bibliography

- [1] A. T. C. T. 9.9, “Data center storage equipment thermal guidelines, issues, and best practices.” [https://resourcecenter.ashrae.org/File%20Library/Technical%20Resources/Bookstore/ASHRAE\\_Storage\\_White\\_Paper\\_2015.pdf](https://resourcecenter.ashrae.org/File%20Library/Technical%20Resources/Bookstore/ASHRAE_Storage_White_Paper_2015.pdf), 2015.
- [2] “Data center cooling best practices.” <https://www.datacenterknowledge.com/data-center-cooling-best-practices/>, Feb 2017.
- [3] *Effect of Relative Humidity, Temperature and Gaseous and Particulate Contaminations on Information Technology Equipment Reliability*, vol. Volume 1: Thermal Management of *International Electronic Packaging Technical Conference and Exhibition*, 07 2015. V001T09A015.
- [4] T. I. Association *et al.*, “Tia-942 telecommunications infrastructure standard for data centers,” *Arlington Virginia: Telecommunications Industry Assn*, 2010.
- [5] T. Saba, A. Rehman, T. Sadad, H. Kolivand, and S. A. Bahaj, “Anomaly-based intrusion detection system for iot networks through deep learning model,” *Computers and Electrical Engineering*, vol. 99, p. 107810, 2022.
- [6] W. Stallings, *Network Security Essentials: Applications and Standards*. William Stallings books on computer and data communications technology, Prentice Hall, third ed., 2007.
- [7] Node.js, “Node.js.” <https://nodejs.org/>.
- [8] P. G. D. Group, “PostgreSQL.” <https://www.postgresql.org/>, Feb. 2023.
- [9] “Prisma | Next-generation ORM for Node.js & TypeScript.” <https://www.prisma.io/>.
- [10] “React – A JavaScript library for building user interfaces.” <https://reactjs.org/>.
- [11] “Next.js by Vercel - The React Framework.” <https://nextjs.org>.
- [12] “Tailwind CSS - Rapidly build modern websites without ever leaving your HTML..” <https://tailwindcss.com/>.